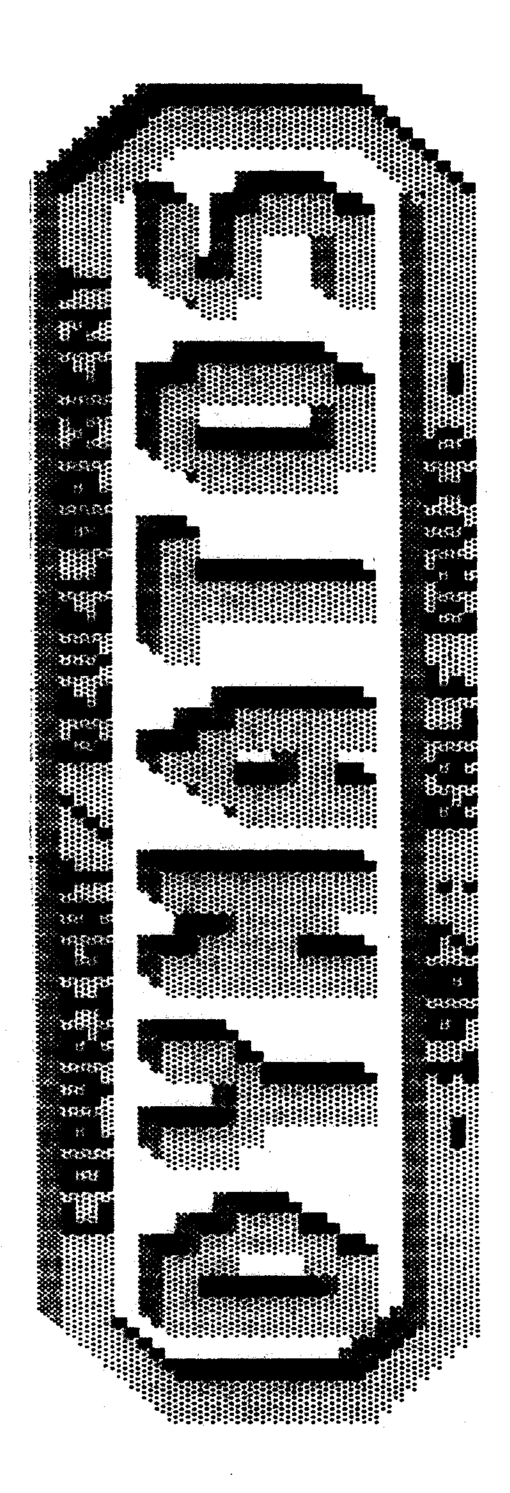
Version 1.6 a



DYNATOS 1.6a (C) 1987 Ralf David

Ein Diskettenmonitor für ATARI XL/XE Computer mit mindestens 64 KByte RAM und Diskettenstation.

ACHTUNG: Hersteller und Verkäufer übernehmen keine juristische Verantwortung und keine Haftung für Schäden jeglicher Art, die direkt oder indirekt durch Besitz, Anwendung oder Mißbrauch von DYNATOS entstehen könnten !

Ladeanweisung :

- Computer und Floppy ausschalten
- Floppy einschalten, DYNATOS-Diskette einlegen & Hebel schliessen
- Computer einschalten

> DYNATOS wird geladen !

DYNATOS wird hautsächlich über die bewährte, leicht bedienbare Menütechnik gesteuert. Einfach eine Taste tippen (eventuell zusammen mit <CONTROL>), und die entsprechende Funktion wird aufgerufen.

Im folgenden werden nun alle Funktionen erklärt :

*** QUIT :

Rücksprung ins Hauptmenü.

*** DIRECTORY :

Das Disketteninhaltsverzeichnis wird auf den Bildschirm ausgegeben. Zusätzlich werden zu jedem File Status (ungesichert / gesichert / gelöscht), Filelänge und Startsektor angegeben. Bis zum ersten Leereintrag werden alle Einträge angezeigt. (Es kann deshalb vorkommen, daß man längst gelöschte Files in der Directory noch wiederfindet)

*** LOAD SECTOR :

Lädt einen anzugebenden Sektor und zeigt ihn in Form eines HEX- und ATASCII- Dumps. Mit der Tabulatortaste (TAB) kann man das ATASCII-Dump auf Bildschirmcode umschalten (und wieder zurück). Das hat sich beim Suchen und Editieren von Texten als extrem nützlich erwiesen.

*** NEXT SECTOR OF FILE :

Lädt orientiert am DOS 2.x Fileformat den logisch nächsten Sektor eines Files bezogen auf den gerade geladenen Sektor. Wenn bereits der letzte Sektor geladen war, gibt es eine Fehlermeldung.

*** PREVIOUS SECTOR :

Lädt den Sektor mit der sequenziell nächstkleineren Nummer.

*** NEXT SECTOR :

Lädt den Sektor mit der sequenziell nächstgrößeren Nummer.

*** SAVE SECTOR :

Schreibt den Sektor, den man gerade als Dump vor sich sieht, auf Disk. Bei der Sicherheitsabfrage hat man hier neben dem üblichen Y(es)/N(o) jedoch noch die Möglichkeit, mit O(ther) die Nummer des Zielsektors zu ändern, wonach dann auch eine erneute Abfrage erfolgt.
Vor dem Schreiben wird noch der Sektor mit entsprechender Nummer von Diskette gelesen und in die Backup-Schieberegisterkette eingetragen (für alle Fälle).
(Mehr dazu -> SECTOR MANAGER !)

*** EDIT SECTOR :

Der Sektoreditor ist wohl die leistungsfähigste Komponente von DYNATOS. Hier kann den Bytes und Bits auf alle möglichen Formen zu Leibe rücken werden :
Zuerst einmal kann man natürlich auch hier das Text-Dump mit TAB in die gewünschte Form bringen, was dann auch oben rechts angezeigt wird.
Dann kann man den gewünschten Code einstellen, in dem editiert werden soll. Man kann hier zwischen 6 verschieden Möglichkeiten wählen :

DEC - Dezimalsystem	Taste: CONTROL-TAB, dann "D"
HEX - Hexadezimalsystem	Taste: CONTROL-TAB, dann "H"
BIN - Dualsystem (binär)	Taste: CONTLOL-TAB, dann "B"
ASC - ATASCII-Code	Taste: CONTROL-TAB, dann "A"
COD - Bildschirmcode	Taste: CONTROL-TAB, dann "C"
ASM - Assembler !!!	Taste: CONTROL-TAB, dann "M"

Alle Zahleneingaben sollten <=255 (FF;11111111) sein. Im Zweifelsfall

wird der LOW-Anteil einer Eingabe verwendet. HEX-Zahlen müssen 2- oder 4-stellig sein, BIN-Zahlen 8-stellig. Außerdem sind die Symbole "$" und "%" überflüssig. Bei ASC und COD wird nicht auf RETURN gewartet, sondern die Eingaben werden sofort übernommen, so daß man fast ganz normal schreiben kann.
Bei "VALUE" wird das mit dem Cursor bestimmte Byte im gerade aktuellen Editiercode dargestellt. Falls ASM eingestellt wurde, wird an der Cursorposition ein Maschinenbefehl disassembliert, und (bis auf 3 kleine Ausnahmen) in der üblichen Standartsyntax gezeigt. Mit SHIFT-TAB kann man den Disassembler von HEX auf DEC (und umgekehrt) umschalten, und mit SHIFT-"*" kann man den Cursor maschinencodegerecht logisch nach rechts bewegen.
Die erwähnten Abweichungen von der Standartsyntax sind durch die Form des Assemblers bedingt. Der (Dis)Assembler muß innerhalb eines festen Codeblocks, nämlich dem Sektor, Maschinenbefehle editieren können, ohne die Position des übrigen Codes zu ändern. Das bedeutet auch, daß der Assembler zB. nicht willkürlich nur anhand des Betrages einer Zahl (zB. LDA 20) entscheiden darf, ob der Befehl 3 Bytes belegen soll, oder er wird Zeropage adressiert, und belegt somit nur noch 2 Bytes. Das Maschinenprogramm könnte hinterher abstürzen oder sonst was für chaotische Dinge fabrizieren, nur weil zB. "LDA 53770" zu "LDA 20" gemacht wurde und jetzt plötlich ein undefinierten Byte im Code umhergeistert. Damit das nicht geschieht, muß man es dem Assembler folgendermaßen mitteilen, wenn er Zeropage adressieren soll :

```
  LDA Z,20   wird zu  A5 14     assembliert  (Zeropage!),
  LDA 20     wird zu  AD 14 00  assembliert !
```

Weiterhin muß man relative Sprünge selbst berechnen (auszählen), weil es in einem Sektor nun mal weder Labels noch Adressen gibt. Die Syntax für relative Sprünge ist "BEQ R,10 ; BPL R,246". Der letzte Punkt ist, daß die Akkumulatoradressierung nicht extra angegeben werden muß, sondern zB. einfach "LSR" anstatt "LSR A" zu schreiben ist.

*** BUFFER OPERATOR :

Diese Funktion ist im Prinzip eine RAMdisk. Die 22 kByte RAM, welche bei den XL/XE-Geräten unter den ROMs liegen, lassen sich nämlich vorzüglich als Kopier/Sicherheits/Arbeits-Buffer verwenden. Bis zu 170 Sektoren kann man in diesem Buffer unterbringen.
Die Arbeit mit dieser primitiven RAMdisk sieht so aus, daß zum Laden und Saven immer 3 Parameter benötigt werden :
1. BUFferposition - gibt an, ab welchem innerhalb der 170 Buffersektoren gelesen oder geschrieben werden soll
2. SEKtor - gibt an, ab welchem Sektor auf der Diskette es losgeht
3. LENght - gibt an, wie viele Sektoren kopiert werden sollen
BUF+LEN-1 sollten dabei natürlich nicht >170 sein.
Um die Arbeit etwas zu erleichtern werden jeweils die Parameter der letzten 5 <LOAD>-Aufrufe (Eintragungen in den Buffer) gemerkt und angezeigt.
Außerdem gibt es noch 2 Funktionen, mit denen man direkten Zugriff auf die Sektoren in der RAMdisk hat :
<EDIT> editiert einen Sektor aus der RAMdisk, und <-BUF> schreibt einen gerade bearbeiteten (EDIT SECTOR oder <EDIT>) oder geladenen Sektor (LOAD SECTOR) in die RAMdisk.
Ein Aufblitzen in der Bildschirmmitte bei Bufferoperationen kommt übrigens daher, daß mit den ROMs auch der Zeichensatz kurz abgeschaltet wird.

*** CLEAR SECTOR :

Setzt alle Bytes des gerade geladenen Sektors auf 0.

*** FIND SEQUENCE :

Sucht in einem anzugebenden Bereich oder File nach einer Sequenz von vorzugebenden Daten (zB. Wörter, Zahlen(ketten), ...). Zuerst legt man fest, ob in einem File oder sequenziell gesucht werden soll. Wenn man ein File absuchen will, muß man den Filename eingeben, für eine sequenzielle Suche müssen entsprechend die Grenzen des Suchfeldes eingegeben werden. Als nächstes gibt man an, nach wie vielen Bytes gesucht werden soll, und bestimmt dann, in welchem Code die Eingabe des Suchwortes erfolgen soll. Zuletzt gibt man dann byteweise das Suchwort ein.
ZB. : Es soll im File "SUPER.BAS" das Wort "FIRE" gesucht werden. Dazu wären folgende Eingaben notwendig :

"F" - FIND SEQUENCE aufrufen
"F" - FILE SEARCH anwählen
"SUPER.BAS"-RETURN - Filename eingeben
"4"-RETURN - "FIRE" hat 4 Buchstaben
"A" - Buchstaben im ATASCII-Code eingeben
"F"-RETURN - "FIRE" wird
"I"-RETURN - Buchstabenweise
"R"-RETURN - nacheinander
"E"-RETURN - eingegeben

Wenn DYNATOS fündig wird, werden Sektornummer und Byteposition ausgegeben und nachgefragt, ob weitergesucht werden soll . Die Suche kann mit ESC auch vorzeitig abgebrochen werden.

*** SECTOR MANAGER :

Eine Funktion, bei der man Zugriff auf die schon erwähnten Backupregister und ein allgemeines Sektorregister (SECTOR BANK) hat.
Immer bevor ein Sektor geschrieben wird (bei SAVE, VTOC & RELINK), wird der entsprechende Sektor von Disk geladen und in die Backupregister eingetragen. Diese 6 Backupregister sind als Schieberegister organisiert, so daß man von den 6 zuletzt überschriebenen Sektoren Backups des Originalzustandes hat. Immer wenn ein neuer Sektor in die Backupregister kommt, fliegt der letzte (#6) aus dem Backupregister raus.
Weiterhin gibt es 9 frei nutzbare Sektorregister, die Sektorbank, die man beliebig als Zwischenspeicher oder für ähnliche Dinge nutzen kann. Mit STORE kann man den vorher zB. mit LOAD geladenen Sektor in eins der Register #A bis #I eintragen. Mit EDIT kann man einen Sektor aus einem beliebigen Register in den Sektor-Editor laden. Die Sektornummer eines Sektors im Register kann man mit NUMBER ändern, und mit COMMENT kann man zu den Registern der Sektorbank noch einen kurzen Kommenter schreiben.

*** SECTOR CODER :

Diese Funktion macht es möglich, beliebige Bytes eines Sektors aus einer beliebigen Sektorsequenz logisch oder aritmetisch mit einer ebenso beliebigen Konstante zu verknüpfen. Dabei kann man zwischen AND, OR, EOR, "+", "-" und "=" wählen.
Zu Anfang muß man zwischen ANALYSE und MODIFY wählen, was grob gesagt nichts anderes heißt, als Code knacken oder selbst (en/de)codieren :

ANALYSE :

Verknüpfung auswählen und angeben, welchen Sektor man sich vornehmen will. Der Sektor wird dann geladen und als Dump auf den Bildschirm gebracht (TAB -> ASC/COD !). Mit "<" und ">" kann man jetzt verschiedene Zahlen durchprobieren wobei synchron dazu das Dump entsprechend umcodiert wird. Mit RETURN kann der Sektor jederzeit abgespeichert werden. ESC bricht die Suche ab.

MODIFY :

Einfach die Verknüpfung wählen, eine Konstante angeben, und dann festlegen, von welchem und bis zu welchen Byte die Verknüpfung pro Sektor durchgeführt werden soll. Zuletzt muß dann noch eingegrenzt werden, welche Sektoren bearbeitet werden sollen.
Will man zB. jeweils die ersten 10 Bytes in den Sektoren 122 bis 125 auf 0 setzen, dann ist folgendes zu tippen :

```
  CONTROL+"C"     - SECTOR CODER aufrufen
  "A"             - AND anwählen
  "0"-RETURN      - Konstante
  "1"-RETURN      - von Byte   1
  "10"-RETURN     - bis Byte  10
  "122"-RETURN    - von Sektor 122
  "125"-RETURN    - bis Sektor 125
```

Mit ESC kann man die Funktion jederzeit stoppen.

*** CALCULATOR :

Ein Minirechner, mit dem man die gebräuchlichsten Rechnungen schnell durchführen kann.
Zwischen 5 Möglichkeiten kann man hier wählen :

1. HEX zu DEC Umwandlung - Operator "$", Hexzahl 2- oder 4-stellig.
2. DEC zu HEX Umwandlung - Operator "!"
3. BIN zu DEC Umwandlung - Operator "%", Binärzahl 8-stellig
4. Aufspaltung einer Dezimalzahl in High- & Lowbyte - Operator "#"
5. Gemischte Rechnung mit den Operatoren +,-,*,/,^,(,) . Es dürfen DEZ-, HEX- & BIN- Zahlen in den Rechnungen verrechnet werden.

Syntaxbeispiele:
$C4 $D301   !1536   %10111011   #53761   2*$F2+%01010101 .

*** PRINT SECTOR :

Druckt den Sektor, den man als Dump vor sich hat, in Form einer Bildschirmkopie ("Kopie" ist nicht ganz wörtlich zu nehmen) aus.

*** PRINTER ON/OFF :

Hier kann man bestimmen, ob bei verschiedenen Funktionen nach einer Druckerausgabe gefragt werden soll. Wer also meint, einen Ausdruck der DIRECTORY, der FIND-Data, des DISK-DUMPs oder sonst was zu brauchen, sollte "PRINTER ON" einstellen.
Die Funktion "PRINT SECTOR" wird davon NICHT beeinflußt.

*** ESC :

Bewirkt den Abbruch einer Funktion.

*** VTOC OPERATOR :

Ein Editor, mit dem man die DOS 2.x Sektorbelegungstabelle (VTOC-Sektor) editieren kann.
Im Status-Display sind jeweils die Statusinformationen von 16 Sektoren aufgeführt. Ganz links steht die Sektornummer, auf die der Cursor positioniert ist.
Mit den Cursortasten kann nun horizontal die 16 Sektorn abfahren und editieren (S/R) und vertikal die 16 vorhergehenden bzw. nachfolgenden Sektorstatusinformationen in das Display schieben, und mit "?" kann man den Cursor auch direkt positionieren. Übrigens kann man zwar immer 1040 Sektoren editieren, auch wenn man eine SD-Disk editiert, aber alles über 720 hat dann natürlich keinen Sinn !

*** FORMAT DISK :

Formatiert eine Diskette in single- oder middle- (enhanced) Density. Sollte die formatierte Diskette für DOS 2.x verwendbar sein, dann muß zusätzlich noch die DOS-Initialisierung vorgenommen werden.

*** RELINK :

Mit dieser Funktion bekommt man die DOS 2.x Sektorverkettung leicht in Griff. Einfach die Sektornummer eingeben, und man bekommt die Verkettungsparameter geliefert, die man daraufhin sofort ändern kann.

*** DISK DUMP :

Erzeugt eine grobe Statustabelle über die einzelnen Sektoren einer Diskette. Es gibt dabei 3 Möglichkeiten :
1. "*" - *normaler* beschriebener Sektor
2. "0" - leerer Sektor (alles nur Nullen)
3. "E" - der Sektor hat einen Fehler (oder er existiert nicht)
Es ist ratsam, sich die Tabelle ausdrucken zu lassen, da sie nicht ganz auf den Bildschirm passt und anfängt zu scrollen.

*** RENUMBER FILE :

Mit dieser Funktion kann die Filenummer, die in den beiden vorletzten Bytes eines jeden DOS 2.x Filesektors codiert ist, im ganzen File geändert werden.
Die Directory wird von DYNATOS jedoch nicht verändert. Gegebenenfalls muß man den Directoryeintrag also von Hand (im Editor natürlich) ändern.

*** CREATE BIN-FILE LOADER :

Schreibt einen für DOS unsichtbaren Mikroloader für "BINARY"-Files auf leere oder auch schon volle Disketten. Nur die 3 Bootsektoren und die letzten beiden Directorysektoren, die sowieso nie benutzt werden, werden vom "BIN-FILE LOADER 2" beansprucht. Also ideal für Programmsammlungen !
Bedienung des erzeugten BIN-FILE LOADERS :
Einfach booten, die Directory abwarten und den dem File zugeordneten Buchstaben tippen, fertig. Nach einem RESET wird (falls der Loader bis dahin nicht rausgeflogen ist) der gesammte Speicher auf 0 gesetzt, (also bereit und ohne störende Einflüsse für neue Programme) und erneut eine Directory geladen.

*** CREATE BOOTABLE BASIC :

Macht aus ganz normalen BASIC-Programmen Bootdisketten. Das hat erstens den Vorteil, daß die leider oftmals unterschätzten BASIC-Programme gleich in ein besseres Licht gerückt werden (Zitat: "BASIC??? öähhh... aber "ne Bootdisk!, doch erst ma' sehn"), und zweitens hat man dadurch wesentlich mehr Platz, weil ja das DOS wegfällt. Nebenbei kann man jetzt sogar schon beim Laden den Titel nennen ("LOADING SUPERPROGRAM"). Wenn man seine Bootdisk erzeugt hat, braucht man nur Sektor 2 zu laden, in den Editor zu gehen, auf COD schalten und die letzten 40 Bytes editieren, welche genau dafür reserviert sind. Wem der voreingestellte Loadsound auf die Nerven geht, der kann sich an Byte 74 (immer noch Sektor 2) versuchen. Eine 0 schaltet den Sound ab.

<u>An den User :</u>

DYNATOS ist ein noch neues Programm. Es kann deshalb sein, daß vielleicht irgentwo noch der Wurm im Code steckt. Deshalb möchte ich jeden User, der einen solchen findet, bitten, mir das mitzuteilen.
Auch wer sonstige Anregungen und Verbesserungsvorschläge hat, möge mir diese doch mitteilen. (Der Speicher des ATARIs ist mit DYNATOS schon 99.9%ig ausgelastet).
Gegebenenfalls gibt es Gratisupdates.

<u>DER AUTOR !</u>

*** DYNATOS *** - Ergänzung zu Version 1.6a - 28.06.1987

DYNATOS 1.6 (C) 1987 Ralf David

RENUMBER FILE :

Mit dieser Funktion kann die Filenummer, die in den beiden vorletzten Bytes eines jeden DOS 2.x Filesektors codiert ist, im ganzen File geändert werden.
Die Directory wird von DYNATOS jedoch nicht verändert. Gegebenenfalls muß man den Directoryeintrag also von Hand (im Editor natürlich) ändern.

BUFFER OPERATOR :

Hier wurden noch die Menüpunkte <EDIT> und <-BUF> implementiert.
<EDIT> editiert einen Sektor aus der "RAMdisk", und <-BUF> schreibt einen gerade bearbeiteten (EDIT SECTOR oder <EDIT>) oder geladenen (LOAD SECTOR) in die "RAMdisk".

EDIT SECTOR :

Kleine Veränderung gegenüber V1.5 :
EDIT SECTOR lädt keine Sektoren mehr ein, sondern editiert einfach alles, was gerade im Sectorbuffer (nicht der von BUFFER OPERATOR !) von DYNATOS ist.
Vorteil : Man kann zB. den Editor zwischendurch kurz verlassen (auch aus Versehen mit ESC). Einfach Editor aufrufen und es geht weiter.

Folgende mir bekannte Fehler der Version 1.5 wurden behoben :
1. Man kann den CALCULATOR jetzt auch mit ESC verlassen, ohne daß die interne Adressenverrechnung abstürzt.
2. Der BUFFER OPERATOR wurde gegen die bisher mögliche (meist fatal endende) Bereichsüberschreitung der Speicherbereiche geschützt.